Un formalisme pour la spécification de langages discontinus et à ordre libre

Sophie Robin, Michèle Courant

INRIA

CENTRE DE RENNES

IRISA

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
BP 105
78153 Le Chesnay Cedex
France
Tél. (1) 39 63 55 11

Rapports de Recherche

N° 544

UN FORMALISME POUR LA SPÉCIFICATION DE LANGAGES DISCONTINUS ET À ORDRE LIBRE

Sophie ROBIN
Michèle COURANT

Juillet 1986

IRISA

INSTITUT DE RECHERCHE EN INFORMATIQUE
ET SYSTÈMES ALÉATOIRES

Campus Universitaire de Beaulieu
Avenue du Général Leclerc
35042 - RENNES CÉDEX
FRANCE
Tél. : (99) 36.20.00
Télex : UNIRISA 95 0473 F

UN FORMALISME POUR LA SPECIFICATION DE LANGAGES DISCONTINUS ET A ORDRE LIBRE

Sophie ROBIN - Michèle COURANT

Juin 1986
Publication Interne n°303
20 pages

RESUME

Une étude du langage des Petites Annonces, a exhibé quant à ce fragment de langage naturel, un certain nombre d'originalités. Les phrases relevant de ce corpus se caractérisent en particulier par l'ordre libre des mots conjugué à la notion de chaîne "pertinente".

Ceci nous a conduits à la définition d'un formalisme de type grammaire hors-contexte attribuée visant à tirer parti des propriétés du langage. Or, bien que définies dans un contexte plus général, les Gapping Grammars répondent à des objectifs similaires.

Cet article se compose successivement d'une présentation puis d'une étude comparative des deux formalismes.

ABSTRACT

A linguistic study of classified advertisement language has pointed out certain original features of this natural language corpus. In particular, the study has showed that the phrase analysis on this corpus requires a combination of free word order expression and of the notion of "pertinent" string.

This study has lead us to design a formalism which is based on attribute context-free grammars and which is aimed at making use of these language properties. Although they were designed for use in a more general context, Gapping Grammars turn out to be useful for studying the same problems.

The present paper describes both formalisms before providing a comparison of them.

SOMMAIRE

1-INTRODUCTION

Dans le contexte du système HAVANE, système-expert noyau dédié à des applications de type mise en relation de Petites Annonces (P.A) exprimées en langage naturel, nous avons entrepris une étude linguistique du corpus concerné.

Cette étude nous a conduits à la définition d'un formalisme de type grammaire attribuée permettant la spécification des connaissances requises notamment pour l'analyse de phrases d'un tel langage.

Pour cette définition, à la fois compte-tenu des propriétés du corpus et des objectifs de l'analyse, les critères retenus ont été les suivants: d'une part l'expression aisée de l'ordre libre des mots dans une phrase, d'autre part la spécification limitée aux seuls éléments textuels "pertinents" des phrases, c'est-à-dire jugés significatifs dans le cadre de l'application. Il s'ensuit bien entendu une certaine adéquation du formalisme défini, à ces deux caractéristiques.

Or le formalisme des Gapping Grammars (GG) défini par ailleurs par V. DAHL et H. ABRAMSON se révèle également être un formalisme très adapté à la définition de langages possédant ces propriétés.

Dans cet article nous présentons dans un premier paragraphe le langage de spécification conçu dans le cadre des P.A. Dans le second paragraphe nous procédons ensuite à une évaluation de ce formalisme, basée sur une étude comparative avec les GG.

2-ETUDE REALISEE SUR LE LANGAGE DES PETITES ANNONCES

2.1-CADRE DE L'ETUDE

Lors de la conception du système HAVANE, nous avons entrepris une étude relative à la représentation des connaissances sous-jacentes à l'analyse et à la génération d'un langage style "petites annonces" [COUR 85],[BOSC 85].

Le but de ce système étant la mise en relation de petites annonces, la première fonctionnalité de l'interface usager est d'assurer la compréhension de phrases rédigées dans le style "P.A". Or dans cette application les énoncés constituent des questions à une banque de données. Un énoncé doit donc conduire à une interprétation normalisée construite à partir d'un ensemble prédéfini d'informations élémentaires significatives au regard des objectifs de l'application. Dès lors le rôle d'un analyseur est d'isoler les éléments textuels pertinents correspondants et

d'élaborer sur cette base, la structure sémantique standard requise.

La seconde caractéristique de l'interface développée est d'être dotée d'une capacité de paraphrasage des compréhensions pour validation de l'interprétation d'une question.

Enfin, dans le cadre spécifique de la mise en relation automatique de P.A, l'interface comporte également un aspect génération de phrases. Ceci provient du fait que dans un tel système, une P.A concrétise à la fois une question à la banque de données et une réponse susceptible d'être fournie lors de consultations ultérieures d'autres usagers. Le système étant capable d'acquérir par dialogue de nouvelles informations, il faut assurer la conformité entre texte et compréhension en répercutant sur la P.A, les informations issues du dialogue. Pour ce faire l'idée est de procéder à une reformulation automatique du texte initial.

Dans ce premier chapitre, après avoir décrit le langage style "P.A", nous nous attachons à présenter les formalismes choisis pour la représentation des connaissances nécessaires à l'analyse, au paraphrasage et à la reformulation pour un tel langage.

2.2-CARACTERISTIQUES DU LANGAGE STYLE "PETITES ANNONCES"

D'un point de vue syntaxique le langage "style P.A" se définit comme un langage souvent construit à partir de formulations abrégées, essentiellement caractérisées par un effacement des mots de liaisons tels que prépositions, conjonctions, articles, pronoms etc. Une phrase de ce langage apparaît donc comme un ensemble d'unités syntaxiques facultatives, indépendantes, et agencées dans un ordre quelconque.

On trouve ainsi des annonces telles que "*vends/ quartier calme/ beau pavillon/ T6/ garage/ jardin 800 m2/ environs Rennes*" dont un grand nombre de variantes équivalentes résultent d'un agencement différent des composants élémentaires. Cette équivalence ne se retrouve pas en revanche dans le cas des deux tournures "*cherche à vendre maison*" et "*cherche maison à vendre*" qui décrivent en réalité des transactions totalement opposées.

En dépit de la simplicité des constituants syntaxiques élémentaires, l'analyse de telles phrases n'est pas triviale. Tout d'abord il importe, comme dans tout langage naturel, de pouvoir interpréter les termes dans leurs contextes. Ainsi dans une P.A immobilière, la chaîne "maison" peut être interprétée comme l'objet de la transaction excepté dans des locutions telles que "*près maison de la culture*". Cette prise en compte des contextes nécessite donc la connaissance de structures

syntaxiques relativement complexes. Par ailleurs les besoins même de l'application peuvent requérir une analyse syntaxique plus élaborée. C'est le cas par exemple lorsque d'une description détaillée du logement telle que *"grande maison sur sous-sol, comprenant cuisine, séjour, sdb, 1er étage 4 chambres, 2ème étage 50 m2 aménageables"*, on veut déduire un nombre de pièces total.

D'un point de vue sémantique, ce langage présente généralement une grande concision. La petite annonce immobilière suivante : *"appt F4, Rennes, 2000f max"* par exemple, sous-entend qu'il s'agit d'une demande de location de par la présence d'un montant représentant un loyer et la tournure employée pour caractériser celui-ci. L'analyse d'un tel langage doit donc s'appuyer sur de grandes capacités d'inférence et par suite sur un grand nombre de connaissances sémantiques.

2.3-REPRESENTATION DES CONNAISSANCES

Ce paragraphe présente les formalismes définis dans le cadre de l'étude menée sur les P.A, et ce successivement pour l'analyse, le paraphrasage et la reformulation.

2.3.1 L'ANALYSE

En ce qui concerne l'analyse, la spécificité du corpus considéré et les objectifs de l'application, nous ont conduits à une spécification des connaissances tirant parti de la localité des constituants syntaxiques et permettant de diminuer la combinatoire à exprimer. Cette spécification se fait ainsi à deux niveaux:

- lexico-syntaxique : description du vocabulaire et des structures syntaxiques de base,

- sémantique : expression de règles d'inférence et de critères d'acceptabilité.

Le formalisme choisi pour la représentation des connaissances est de type grammaire hors-contexte attribuée. En effet, outre l'adéquation des grammaires attribuées à l'analyse syntaxique, le niveau sémantique se résumant essentiellement à l'expression de règles d'inférence et de cohérence, le calcul d'attributs se révèle également très adapté à ce niveau.

Dans un souci d'uniformisation, le formalisme choisi est le même pour les deux niveaux.

Par ailleurs, dans la mesure où la sémantique globale d'une annonce obéit au principe de compositionnalité (ie est une fonction de la sémantique des constituants élémentaires), seuls les attributs synthétisés ont été retenus au niveau sémantique.

Cette restriction ne réduit en rien la puissance d'expression du formalisme et rend la spécification plus claire.

En conclusion pour un domaine donné, la spécification se traduit par une grammaire appelée Descripteur de Structure (DS), décrivant la sémantique du domaine, et par des grammaires dites Grammaires d'Analyse Syntaxique (GAS) décrivant chacune une unité syntaxique élémentaire.

a)Le Descripteur de Structure

C'est une grammaire telle que :

- autour de son axiome "annonce", sont regroupés sous forme d'attributs toutes les informations requises pour la comparaison d'annonces,
- ses terminaux sont les axiomes des GAS,
- à chacune de ses règles sont associées des expressions de calcul d'attributs et de contrôle sur les valeurs d'attributs,
- la juxtaposition de deux non-terminaux en partie droite d'un règle n'induit aucune contrainte d'ordre et de contiguité sur les chaînes reconnues par ces non-terminaux.

Soient par exemple deux règles rédigées pour la spécification du domaine immobilier:

Règle 1:

<annonce(ST,MT,NAT,N1,N2,...)> --> <transaction(ST1,MT1)>
<objet(NAT1,NBP1,NBP2,...)><échange(...)>

avec

ST1=inconnu et MT1=achat-vente => ST=offre
ST1=inconnu et (MT1=location ou NBP1#NBP2) => ST=demande
défaut(ST):ST1
...
non(NBP1#NBP2 et ST1=offre)
non(NAT1=inconnu)

Les trois premières expressions attachées à cette règle expriment de l'inférence à travers du calcul d'attributs. La première indique qu'en l'absence d'information sur le sens de la transaction (ST1=inconnu) et si le mode de la transaction est de type "achat-vente" (MT1=achat-vente), on peut déduire qu'il s'agit d'une offre (ST=offre). Les deux dernières expressions expriment des contrôles d'acceptabilité. La première signifie que dans le cas d'une transaction de type "offre" (ST1=offre), il

est interdit d'indiquer le nombre de pièces sous forme d'un intervalle (NBP1#NBP2). La seconde exprime que la "nature de l'objet" est une information obligatoire (NON(NAT1=inconnu)).

Règle 2:

<objet(...)> --> <<nature(...)>> <caractéristiques(...)>

avec NAT=studio => N1=1 et N2=1
...
non((NAT=maisonnette) et (NB1>=3 ou NB2>=3))

Dans cette dernière règle, la présence de doubles chevrons autour de "nature" indique qu'il s'agit d'un terminal de la grammaire.

b)Les Grammaires d'Analyse Syntaxique

Elles définissent les structures syntaxiques de base et leur associent une sémantique élémentaire sous forme d'attributs.

Une GAS correspond à une information élémentaire apparaîssant sous la forme d'une chaîne continue dans une phrase. Lorsqu'une information élémentaire peut apparaître sous forme de sous-chaînes discontinues (c'est le cas pour l'information "nature de la transaction" dans des formulations telles que "*cherche maison à louer*"), il est nécessaire de définir plusieurs GAS.

Une GAS est une grammaire telle que :

- ses terminaux sont des chaînes pertinentes rencontrées dans les annonces,
- à chacun de ses non-terminaux peuvent être associées des expressions de calcul d'attributs,
- la juxtaposition de deux non-terminaux en partie droite de règle induit l'ordre et la contiguité des sous-chaînes dérivées de ces non-terminaux dans la phrase.

Exemple

<<nbpièces(X,X)>> --> <<nombre(X)>> pièces | <<type>> <<nombre(X)>>
<<nbpièces(X,X)>> --> F <<nombre(X1)>> bis
avec X=X1+1

...
<<type>> --> type| T | F

2.3.2-LE PARAPHRASAGE

En ce qui concerne la spécification du paraphrasage, celle-ci s'effectue également au travers de calculs d'attributs, à la fois dans le Descripteur de Structure et dans les GAS. Dans la mise en oeuvre actuelle, l'idée retenue a été d'associer à chaque attribut (ou groupe d'attributs) représentant une unité sémantique, dans le DS comme dans les GAS, un attribut de paraphrasage correspondant à l'unité de paraphrase associée.

Les quelques exemples suivants empruntés au domaine immobilier en illustrent bien le mécanisme.

```
<<nature((maison,"une maison"))>> -> maison | pavillon

...

<<nbpièces(((X,X),PHNBP))>>
                        -> <<nombre((X,PHX))>> pièces
                           avec PHNBP="comprenant "+PHX+" pièces"

                           |type <<nombre((X,PHX))>>
                           avec ...

...

<annonce((ST,PHST),(MT,PHMT),((N1,N2),PHN1N2),...)>
                        --> <transaction((ST1,PHST1),(MT1,PHMT1))>
                            <objet((NAT1,PHNAT1),...)>
                            <échange(...)>

    avec
    ST1=inconnu et MT1=achat-vente => ST=offre et
                                      PHST="vous offrez"
     ...
     défaut((ST,PHST)):  (ST1,PHST1)
     ...
```

2.3.3-LA REFORMULATION

Dans l'application "Petites Annonces" considérée, la possibilité de mise en oeuvre d'un dialogue relatif au paraphrasage a pour but d'aboutir rapidement à un accord entre usager et système.

Ce dialogue induit pour des raisons de conformité entre texte initial et compréhension, la problématique nouvelle de la reformulation automatique de phrases "style P.A" [VIBE 84]. Celle-ci consiste en la suppression, l'ajout et la modification d'éléments textuels, sur la base d'une sémantique agréée par usager et système. Aussi pour éviter l'écueil de mises-à-jour

aberrantes du texte, une connaissance approfondie des structures syntaxiques s'avère nécessaire. Cette connaissance doit notamment permettre de relier les chaînes décrites par les GAS à leurs contextes, à savoir permettre par exemple de rattacher des adjectifs a priori non significatifs, aux noms etc.

Pour ce faire, en accord avec les choix effectués quant à l'analyse, nous avons opté là encore pour une solution locale, visant à épargner la description exhaustive du langage.

En s'appuyant également sur un formalisme de type grammaire hors-contexte, l'idée est de spécifier à l'aide de grammaires dites "de contexte", les contextes gauche et droit de chacune des GAS.

Ceux-ci correspondent grossièrement:
- pour un verbe : au groupe sujet, aux adverbes et au groupe complément s'y rapportant,
- pour un nom: à l'article, aux groupes adjectifs, au complément de nom,
- pour un adjectif: aux adverbes le modifiant.

Les contextes étant ainsi définis, il apparaît dans le langage des P.A, une certaine discontinuité syntaxique des éléments contextuels, à la fois entre eux et relativement à la chaîne de référence décrite par la GAS.

Soit le cas de la GAS "*nature*" (analysant ici la chaîne "*maison*") dans la petite annonce suivante :

"Cherche à louer maison, Rennes, située quartier calme, 5 pièces min, meublée ou non, urgent"

GAS "nature" Eléments de contexte

D'où l'idée d'introduire la notion de "bruit" dans les grammaires de contextes (un "bruit" étant par définition une chaîne quelconque de caractères), ceci par le biais d'un non-terminal <<bruit>> se dérivant dans une chaîne terminale quelconque.

Pour la GAS "*nature*", une grammaire simplifiée décrivant son contexte gauche peut ainsi s'écrire :

<<nature-cg>> ---> <<bruit>><<article>><<groupe-adjectif>>
| <<bruit>><<groupe-adjectif>>| ε
<<groupe-adjectif>> ---> <<adjectif>> <<suite-adjectif>>
<<suite-adjectif>> ---> et<<suite-adjectif>>
|ou<<suite-adjectif>> | ε

où <<*article*>> et <<*adjectif*>> sont des entrées lexicales.

3-EVALUATION DES FORMALISMES PROPOSES

3.1-INTRODUCTION

Les formalismes que nous avons définis dans le cadre du système HAVANE, sont basés sur les grammaires attribuées. Ils s'apparentent ainsi aux grammaires logiques [DAHL 83].

Rappelons qu'une grammaire logique est une grammaire de type 0 dans laquelle les symboles sont augmentés d'arguments (ou attributs) et où des procédures peuvent être invoquées en partie droite de règle (ceci pour exprimer des contraintes d'acceptabilité). De plus du calcul d'attributs peut être spécifié au travers du mécanisme d'unification.

Depuis l'introduction des grammaires logiques par A. Colmerauer [COLM 78], de nombreuses variantes ont été proposées, motivées parfois par leur simplicité de mise en oeuvre (Definite Clause Grammars [PERE 80]), mais le plus souvent par une amélioration du pouvoir d'expression (Extraposition Grammars [PERE 81]) et l'adéquation à la description de phénomènes linguistiques particuliers (Definite Clause Translation Grammars [ABRA 84]).

Le formalisme le plus général à l'heure actuelle semble être celui des Gapping Grammars -en français "Grammaires Discontinues"- [DAH 84a]. Ce formalisme se révèle en particulier tout à fait adapté au traitement de l'ordre de mots libre. Ce problème s'étant situé au centre de nos préoccupations lors de l'étude relative aux P.A, nous faisons dans ce deuxième chapitre une étude comparative des formalismes présentés dans le chapitre précédent, et des Grammaires Discontinues.

3.2-LES GRAMMAIRES DISCONTINUES

Le formalisme des Grammaires Discontinues [DAH 84a],[DAH 84b] consiste en une généralisation des grammaires d'extraposition (XG) introduites par F. PEREIRA [PERE 81]. Il s'agit de grammaires logiques dont les règles font référence à des "trous" ("gaps") représentant des sous-chaînes terminales quelconques. Une grammaire discontinue se caractérise par des règles de production de la forme :

$$A, A_1, gap(x_1), A_2, gap(x_2) \ldots A_{n-1}, gap(x_{n-1}), A_n \dashrightarrow B$$

avec: $A \in V_N$

$A_i \in (V_N \cup V_T)^*$ pour i=1,...,n

$B \in (V_N \cup V_T \cup \Gamma)^*$ où $\Gamma = \{gap(x_1), \ldots, gap(x_{n-1})\}$

avec $x_i \in V_T^*$.

Une telle structure de règles de production confère aux GG, outre la puissance d'une machine de Turing, la capacité de désigner à travers les trous des sous-chaînes terminales non spécifiées et de les distribuer dans la partie droite des règles (B) dans un ordre quelconque.

Les GG s'avèrent ainsi adaptées au traitement de divers phénomènes de langue naturelle, en particulier l'extraposition droite, mais surtout l'ordre de mots libre, qui rejoint plus précisément nos propres préoccupations.

Nous illustrons maintenant à travers quelques exemples les possibilités des GG.

Exemple 1: De l'intérêt des trous pour exprimer l'ordre libre

Soit la GG suivante :

A,gap(x),B,gap(y),C --> gap(y),C,B,gap(x)

* Sur la chaîne AEFBDC avec x = EF et y = D, la règle nous donne : A,E,F,B,D,C --> D,C,B,E,F

* Sur la chaîne ABDEFC avec x = [] et y = DEF, la règle nous donne : A,B,D,E,F,C --> D,E,F,C,B

Les GG offrent donc grâce aux trous, un outil puissant - une sorte de méta-grammaire - permettant de décrire globalement une infinité de chaînes, en passant outre les sous-chaînes dont l'analyse n'est pas a priori nécessaire.

Ceci induit la possibilité d'exprimer l'équivalence entre plusieurs chaînes terminales résultant de la combinaison de sous-chaînes terminales élémentaires.

De ce fait l'utilisation des GG s'avère très intéressante en cas de non pertinence de certaines sous-chaînes au regard des objectifs de l'analyse, comme c'est le cas dans l'application P.A. Cette non pertinence se retrouve en outre dans le processus de focalisation susceptible d'être mis en oeuvre en compréhension automatique, processus qui conjugue les avantages: d'une part de l'efficacité de l'analyseur, d'autre part du "réalisme" dans la modélisation d'un interlocuteur au sein d'un dialogue homme-machine.

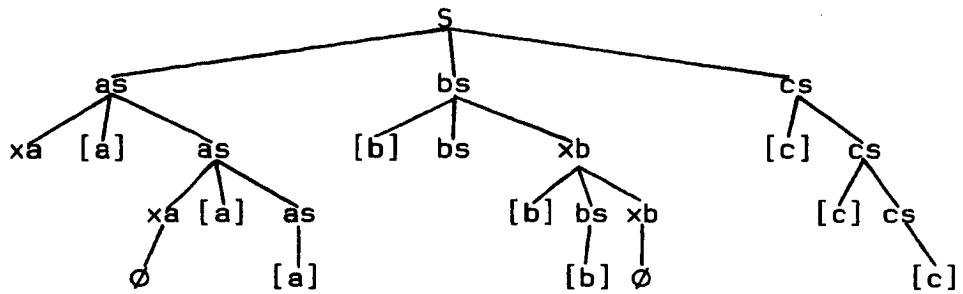
Exemple 2: Puissance et lisibilité

Soit la GG suivante décrivant $a^n b^n c^n$ (n>0):

S --> as,bs,cs
as --> [a] | xa,[a],as
bs --> [b]

xa,gap(x),bs --> gap(x),[b],bs,xb
cs --> [c]
xb,gap(x),cs --> gap(x),[c],cs

L'arbre syntaxique ci-dessous correspondant à $a^2b^2c^2$, illustre bien le fonctionnement de la grammaire, basé sur l'extraposition droite :



Sur un tel exemple, la lecture de [DAHL 84a] nous convainc aisément de la plus-value de lisibilité offerte par les GG par rapport à des formalismes antérieurs tels les XG.

Néanmoins une limite demeure dans le pouvoir d'expression offert par les "gaps". Des langages tels que les langages parenthésés par exemple l'illustrent assez bien. Les trous pouvant en effet désigner des chaînes quelconques, il est de fait impossible de garantir la non absorption des parenthèses par les trous.

Ainsi la GG suivante :

gauche,gap(X),[')'] --> ['('], gap(X).
S --> gauche,[')'],gap(X),S.
S --> [].

reconnaît des expressions bien parenthésées telles que *((a+b)*c)* mais également *((a+b)...*

3.3-COMPARAISON DES FORMALISMES DS+GAS ET GG

3.3.1-POUR L'ANALYSE

L'étude comparative effectuée s'appuie sur trois points de vue, examinés successivement: la puissance d'expression et l'adéquation, d'une part aux langages "discontinus", d'autre part aux langages à ordre de mots libre.

a) Puissance d'expression

Tout d'abord d'un point de vue puissance d'expression, les formalismes DS+GAS et GG - comme la plupart des grammaires logiques - correspondent tous deux à une machine de Turing.

Néanmoins l'utilisation de tels formalismes se révèle plus ou moins aisée -en conséquence ils s'avèrent plus ou moins lisibles - selon les phénomènes à décrire [POPO 85].

Ainsi la description de $a^n b^n c^n$ (n>0) en DS+GAS, gagne-t-elle en simplicité par rapport aux GG:

<<S>> --> <<A(N)>> <<B(N)>> <<C(N)>>
<<A(N1)>> --> a <<A(N)>>
avec N1=N+1
<<A(1)>> --> a
<<B(N1)>> --> b <<B(N)>>
avec N1=N+1
<<B(1)>> --> b
<<C(N1)>> --> c <<C(N)>>
avec N1=N+1
<<C(1)>> --> c

b) Expression de langages discontinus

En ce qui concerne la spécification de langages discontinus, les deux formalismes se révèlent d'une simplicité équivalente. Il convient toutefois de mentionner la plus grande souplesse des GG dans lesquelles les discontinuités peuvent apparaître à tous les niveaux. Le formalisme DS+GAS en revanche, contraint au partionnement des non-terminaux entre DS et GAS selon l'existence d'une discontinuité entre eux ou non.

Ainsi pour décrire une structure telle que :

< A >< B >< trou >< C >< D >< trou >< E >

Le formalisme DS+GAS oblige à regrouper les non-terminaux < A > et < B > ainsi que < C > et < D > sous forme de terminaux du DS afin d'exprimer la contiguité des chaînes qui en dérivent, et ceci en dépit de la décomposition "naturelle" susceptible d'être effectuée.

Ceci nous conduit ainsi à écrire :

1) au niveau DS

<S> -> <<X>> <<Y>> <<E>>

où <<X>> et <<Y>> sont les terminaux introduits pour regrouper <A> et <B> d'une part, <C> et <D> d'autre part.

2) au niveau GAS

<<X>> -> <<A>> <<B>>
<<Y>> -> <<C>> <<D>>
...

En contrepartie, l'expression de trous doit être explicitée dans les GG tandis qu'elle est implicite dans le formalisme du DS, autrement dit n'apparaît dans le DS que ce qui est pertinent au regard des objectifs de l'analyse. Il s'ensuit bien entendu un gain notable en concision, qui compense de fait l'inconvénient évoqué précédemment.

c) Expression de l'ordre de mots libre

En ce qui concerne l'ordre de mots libre, les GG en autorisent la description par l'expression de règles réécrivant un ensemble de combinaisons en une forme équivalente "standard".

La description de phrases élémentaires du Latin peut alors s'exprimer par la grammaire discontinue suivante, extraite de [DAHL 84b]:

```
phrase --> groupe-nominal(nom),groupe-nominal(acc),verbe.
groupe-nominal(Cas) --> adjectif(Cas), nom(Cas).
nom(Cas), gap(G) --> gap(G), [Mot], {dict(nom(Cas),Mot)}.
adjectif(Cas), gap(G) --> gap(G), [Mot],
                          {dict(adjectif(Cas),Mot)}.
verbe, gap(G) --> gap(G), [Mot], {dict(verbe,Mot)}.
dict(verbe,amat).
dict(nom(acc),puerum).
dict(nom(nom),puella).
dict(adjectif(acc),parvum).
dict(adjectif(nom),bona).
```

Une telle grammaire permet ainsi d'analyser des phrases équivalentes issues de la combinaison des éléments:

/ puella / bona / puerum / parvum / amat /.

Dans le DS, de même que la discontinuité, l'ordre de mots libre est implicite. Ce formalisme ayant en effet été défini dans le cadre d'une étude du corpus des P.A, dans lequel la combinatoire et le bruit se trouvent intimement liés, notre choix s'est porté sur un formalisme permettant l'expression conjuguée de l'ordre libre et des trous.

Ce choix se heurte néanmoins à certaines limites. L'utilisation des possibilités du DS pour traduire de façon concise la combinatoire sous-jacente aux phrases du Latin induit ainsi par exemple la grammaire suivante :

<S> --> <groupe-nominal(nom)> <groupe-nominal(acc)>
<<verbe>>
<groupe-nominal(CAS)> --> <<adjectif(CAS)>> <<nom(CAS)>>
<<adjectif(nom)>> --> bona
<<adjectif(acc)>> --> parvum
<<nom(nom)>> --> puella
<<nom(acc)>> --> puerum
<<verbe>> --> amat

Cependant cette grammaire autorise également la reconnaissance de phrases contenant des sous-chaînes parasites absorbées par les trous existant implicitement entre deux symboles de la grammaire du DS.

Si notre choix se justifie donc parfaitement dans le cas des P.A., il convient peut-être pour pallier cet inconvénient, d'étendre le pouvoir d'expression des GAS de façon à permettre aussi au niveau syntaxique l'expression concise de la combinatoire.

Pour ce faire l'idée est de recourir à l'introduction dans les parties droites de règles, d'opérateurs agissant sur les symboles grammaticaux. Une idée similaire a été développée dans [PULL 82].

d) Contraintes sur les trous

Comme nous l'avons vu précédemment, la principale limitation des grammaires discontinues réside dans le fait qu'un trou représente une sous-chaîne terminale quelconque et qu'en conséquence des chaînes non correctes sont reconnues.

L'idée émise par V. DAHL dans [DAHL 84a] pour passer outre cette limitation est:
- soit de spécifier des restrictions sur les trous,
- soit de filtrer les chaînes interdites dans les trous, ceci éventuellement par un processus distinct de l'analyse effectuée par les GG.

Nous avons du reste envisagé des solutions de ce type, lorsqu'il s'est agi d'exclure la reconnaissance du terme "*maison*" dans un contexte tel que "*près maison...* ".
Une solution partielle à ce problème, empruntée dans la mise en oeuvre actuelle, est de définir une GAS "*proximité*" reconnaissant des chaînes telles que "*près maison de la Culture*". Le succès étant accordé à la reconnaissance de la chaîne la plus longue, la définition de la GAS "*proximité*" évite ainsi le succès de la GAS "*nature*" sur la sous-chaîne "*maison*".

Une autre façon de faire serait de caractériser partiellement les langages complémentaires de ceux engendrés par les GAS. Cette approche, tout comme la première solution, rejoint les solutions de type filtre citées dans [DAH 84a] et [RADF 81]. Cette deuxième approche est évidemment beaucoup plus naturelle que l'approche actuelle. En effet, plutôt que de définir une GAS "proximité", il semble plus propre et plus naturel d'écrire :

<<nature(interdit,...)>> -> près maison / proximité maison / ...

3.3.2-POUR LE PARAPHRASAGE ET LA REFORMULATION

En ce qui concerne l'aspect paraphrasage, notre formalisme et celui des GG s'avèrent difficilement comparables. En effet, les GG ne semblent pas avoir été conçues dans un objectif d'adéquation au paraphrasage.

Il nous apparaît malgré tout que l'idée sous-jacente aux GG qui consiste à réécrire un ensemble de combinaisons équivalentes en une structure standard devrait permettre aisément leur utilisation dans le cadre du paraphrasage. Cette idée de structure standard existe d'ailleurs également dans notre formalisme. En effet, la grammaire du DS permet la définition d'une structure sémantique unique pour des phrases équivalentes résultant d'un agencement différent des constituants élémentaires. Cependant cette structure standard n'est à l'heure actuelle pas utilisée à des fins de paraphrasage. Le formalisme devant en effet offrir un moyen de paraphraser les inférences, notre choix s'est porté sur une spécification de la paraphrase parallèle à celles-ci. Comme nous l'avons vu au paragraphe 2.3.2, l'élaboration de la paraphrase s'effectue donc entièrement au travers du calcul d'attributs.

Sur l'aspect reformulation, la comparaison entre les deux formalismes s'avère encore plus délicate. La reformulation de textes, telle que nous l'avons définie, constitue en effet un problème inhérent à l'application Petites Annonces. Le formalisme que nous avons conçu à cette fin se greffe à celui de l'analyse en permettant la description des contextes de chacune des GAS. Malgré la moindre originalité du formalisme à ce niveau, il nous semble intéressant de mentionner l'intérêt de la démarche suivie, à savoir le fait de scinder la spécification en plusieurs étapes. Le gain de simplicité, notamment offert par une décomposition des problèmes, nous paraît en effet devoir rester un critère déterminant pour la définition de formalismes.

4-CONCLUSION

Notre comparaison des deux formalismes GG et DS+GAS s'appuie sur la plus ou moins grande aisance avec laquelle certains phénomènes linguistiques peuvent être décrits.

Ainsi malgré leur adéquation à l'extraposition droite, notamment par rapport à d'autres grammaires logiques, les GG se révèlent d'un usage relativement difficile. Ceci est dû en particulier à la présence de gaps en partie gauche et donc au type 0 des règles de production.

Un formalisme de type grammaire hors-contexte en revanche, s'avère d'un emploi beaucoup plus simple, tandis que l'adjonction d'attributs lui confère la puissance d'expression nécessaire. De ce fait les problèmes restent distincts: d'une part la description de structures (syntaxiques ou sémantiques), d'autre part l'expression des contraintes portant sur cette structure.

La généralisation du langage naturel dans le dialogue homme-machine passant par la spécification de sous-ensembles du langage naturel liés à des applications, il nous paraît essentiel de mettre l'accent sur la souplesse et la lisibilité des formalismes employés.

De cette étude comparative se dégagent ainsi deux résultats relatifs au formalisme DS+GAS:

- tout d'abord, suite à la mise en évidence de certaines lacunes, l'ébauche de quelques solutions basées notamment sur des réflexions de V. Dahl relativement aux GG,

- enfin le renforcement de cette conviction selon laquelle le pouvoir d'expression d'un formalisme repose à la fois sur la concision qui en découle mais également sur sa simplicité d'utilisation, ce qui suppose autant que possible une bonne décomposition des problèmes.

BIBLIOGRAPHIE

[ABRA 84] H. Abramson, "Definite Clause Translation Grammars", Proceedings IEEE Logic Programming Symposium, 6-9 February 84.

[BOSC 85] P. Bosc, M. Courant, S. Robin, "Spécification de connaissances pour une interface en langage à grande liberté syntaxique", 5ème Congrès Afcet RF & IA, Grenoble, Nov 85.

[COLM 78] A. Colmerauer, "Metamorphosis Grammars", Natural Language Communication with Computers, Lecture Notes in Computer Science 63, Springer 1978.

[COUR 85] M. Courant, S. Robin, "Classified advertisement analysis in the context of an expert system in ad matching", Natural Language Understanding an Logic Programming, North-Holland 85, p. 33-47.

[DAHL 83] V. Dahl, "Current trends on Logic Grammars", Proceedings of the Logic Programming Workshop, Albufera (Portugal), 83.

[DAH 84a] V. Dahl, H. Abramson, "On Gapping Grammars", Proceedings of 2nd International Joint Conference on Logic, University of Uppsala, Sweden 1984.

[DAH 84b] V. Dahl, "More on gapping grammars", Proceedings International Conference on Fifth Generation Computer Systems, INGCT, Tokyo 84.

[PERE 80] F. Pereira, D. Warren, "Definite Clause Grammars for Language Analysis", Artificial Intelligence, vol 13, pp 231-278, 1980.

[PERE 81] F. Pereira, "Extraposition Grammars", American Journal of Computational Linguistics, vol 7 no 4, 1981, pp 243-255.

[POPO 85] F. Popowich, "Unrestricted Gapping Grammars: Theory, Implementations and Applications", Thesis Simon Fraser University, July 85.

[PULL 82] G. K. Pullum, "Free word order and phrase structure rules", in J. Pustejovsky and P. Sells eds, Proceedings of the Twelfth Annual Meeting of the North Eastern Linguistic Society, p209-220, 1982.

[RADF 81] A. Radford, "Transformationnal syntax", Campbridge University Press, 81.

[VIBE 84] B. Vibet, "la reformulation automatique des petites annonces dans le système HAVANE" , Rapport de DEA, Université Rennesl, Juin 84.

Imprimé en France

par

l'Institut National de Recherche en Informatique et en Automatique